УДК 004.89

Е.А. Черкашин [1,2], В.А. Попова [2]

[1] Институт динамики систем и теории управления им. В.М. Матросова СО РАН, г. Иркутск, Россия
[2] Институт математики и информационных технологий, Иркутский государственный университет, г Иркутск, Россия

РАСПРЕДЕЛЕННАЯ ИНФРАСТРУКТУРА ДЛЯ ОБРАБОТКИ ДОКУМЕНТОВ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА, ОСНОВАННАЯ НА ГРАФАХ ЗНАНИЙ

Аннотация. В статье рассматривается применение авторских инфраструктурных компонентов, базирующихся на представлении данных в графе знаний и их обработке, основанной на формализованных знаниях. При помощи компонентного проектирования создана среда обработки документов курса университета, включающая их воспроизведение из PDF, хранение, воссоздание на основе хранимых данных. Накапливаемая в графе знаний информация формирует платформу для автоматизации организации образовательного процесса. Основной целью НИРОКР является разработка алгоритмов и программного обеспечения интеграции статических данных с сайта университета, представленных в виде рабочих программ дисциплин, с информационной инфраструктурой университета, например, библиотекой, существующими системами планирования процессов, ранее разработанными магистрантами и профессорско-преподавательским составом подразделений университета.

Ключевые слова: граф знаний, логический вывод, автоматизация образовательного процесса, распределенная обработка данных, автоматизация создания документов

E. A. Cherkashin [1,2], V. A. Popova [2]

[1] Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia
[2] Institute of Mathematics and Information Technologies, Irkutsk State University, Irkutsk, Russia

KNOWLEDGE GRAPH BASED DISTRIBUTED INFRASTRUCTURE FOR PROCESSING EDUCATION PROCESS DOCUMENTS

Abstract. The article deals with the application of the author's infrastructure components based on the representation of data in the knowledge graph and its rule-based processing. The components are used to create an environment for processing university course documents, including their reconstruction from PDF, storage, authoring based on the stored data. The information accumulated in the knowledge graph forms a platform for the automation of the educational process. The main goal of the R&D is to develop algorithms and software to integrate static data from the university website presented in the form of working programs of disciplines with the university information infrastructure, such as library, existing process planning systems previously developed undergraduates and faculty of the university departments.

Keywords: knowledge graph, logical inference, education process automation, distributed data processing, document authoring automation.

Введение. Университет представляет собой сложную социотехническую систему [1], включающую различные компоненты, функционирование которых обеспечивается персоналом с использованием программного обеспечения. Иркутский государственный университет (ИГУ) имеет достаточный (требуемый) уровень автоматизации в областях бухгалтерского учета, планирования образовательного процесса, управления образовательным процессом, оценки успехов обучения студентов, доступа к библиотечным данным. Однако автоматизация имеет островной характер как в аспекте реализованных функций предметной области, так и структурно-организационном аспекте. Институты ИГУ разрабатывают программное обеспечение для решения своих текущих технических проблем и, как правило, не делятся результатами между институтами. Создается впечатление, что просто не существует такой традиции. Решения, имеющие комплексный характер, реализуются специальным отделом Института математики и информационных технологий (ИМИТ) по договоренности. Например, во время пандемии COVID-19 ИМИТ поддерживал функционирование сервера Big Blue Button для всех подразделений ИГУ, спроектировал и внедрил модули Moodle для удаленной регистрации абитуриентов.

Для того чтобы обеспечить выполнение постоянно растущих требований, предъявляемым к функционированию университета, необходимо устранить очевидные недостатки автоматизации учебного процесса. Одной из сложных проблем является подготовка рабочих программ дисциплин (РПД), например, согласование содержимого РПД с учебным планом (УП) конкретной студенческой группы. В РПД, в случае сокращения или изменения количества зачетных единиц (часов), отводимых на лекционные и лабораторные работы, преподаватель должен перепланировать зачетные единицы путем добавления/удаления тем или сокращения/расширения их содержания. Формат представления учебных планов меняется примерно каждые два года, и даже статическую часть содержания шаблона РПД приходится перестраивать. Еще одна задача – книгообеспеченность, проверка ресурсов университетской библиотеки по предоставлению печатных изданий в РПД, проверка и обновление URL-ссылок на электронную литературу. Третья проблема связана с управлением образовательным процессом. Отсутствует общая система планирования нагрузки профессорско-преподавательского состава (ППС), планирование расписаний занятий, мониторинга успеваемости студентов, а именно, факта посещения занятий и полученных оценок.

Вышеупомянутые проблемы решаются преподавателями, как правило, вручную с использованием программного обеспечения автоматизации делопроизводства (Microsoft Office и LibreOffice). Руководство института разрабатывает рекомендации по подготовке РДП, шаблоны документов. Все это призвано поддержать деятельность преподавателей и обеспечить своевременную разработку и обновление РПД: создание документов требуемого качества (во всех мыслимых аспектах). На практике требования качества достаточно трудно удовлетворить из-за ряда факторов. Частично это связано с нынешней недооценкой роли и функций преподавателя в учебном процессе, а также экономическими причинами: преподаватели часто работают в других учреждениях (институтах, вузах, фирмах), и работа с документами не является их основным приоритетом в организации рабочего времени. Создание новых инструментов поддержки деятельности преподавателя, значительно сокращающих трудоемкость подготовки РПД является актуальной задачей, качественное решение которой возможно реализовать с использованием средств искусственного интеллекта (автоматизации обработки естественного языка) и созданием среды интеграции данных из различных источников.

Согласно законодательству РФ сайт ИГУ включает разделы, предоставляющие студентам и преподавателям документы об учебном процессе, включая УП и РПД по всем учебным программам и группам студентов за несколько лет. Документы опубликованы в формате PDF и представляют собой источники содержательных данных об организации учебного процесса. Другая полезная информация опубликована в документах на сайтах Правительства РФ, в том числе справочниками по специальностям, требованиям к специальностям и т.п.

Целью НИРОКР, представленного в данной статье, является автоматизированная поддержка решения творческих задач ППС в части ежегодной подготовке документации об учебном процессе, организации учебных процессов, мониторинг и контроль успеваемости студентов, а в дальнейшем и формирование основы моделирования учебного процесса, проверку публикуемой информации на соответствие стандартам и индивидуальным траекториям обучения студентов.

Используемая платформа НИРОКР. Основой нашей среды интегрирование является распределенное хранилище семантических данных – распределенный графы знаний (ГЗ) [2], реализованный при помощи систем Virtuoso [3], ClioPatria [4], а также адаптаций реляционных баз данных к технологиям Семантического Веба (СВ) [5] (Рисунок 1). Использование распределенных ГЗ позволяет разрабатывать программные подсистемы подразделениям университета достаточно независимо друг от друга, интегрируя подсистемы в рамках общей формализованной модели предметной области в локальной вычислительной сети и Интернет. Модель предметной области основывается на современных достижениях в области разработки Семантического Веба – стандартизованных онтологиях предметных областей и су-

ществующих ресурсах Интернет. Структуры ГЗ в качестве хранилища данных позволяет разработчику создавать программные комплексы как набор взаимодействующих агентов, обменивающихся информацией через содержимое ГЗ.

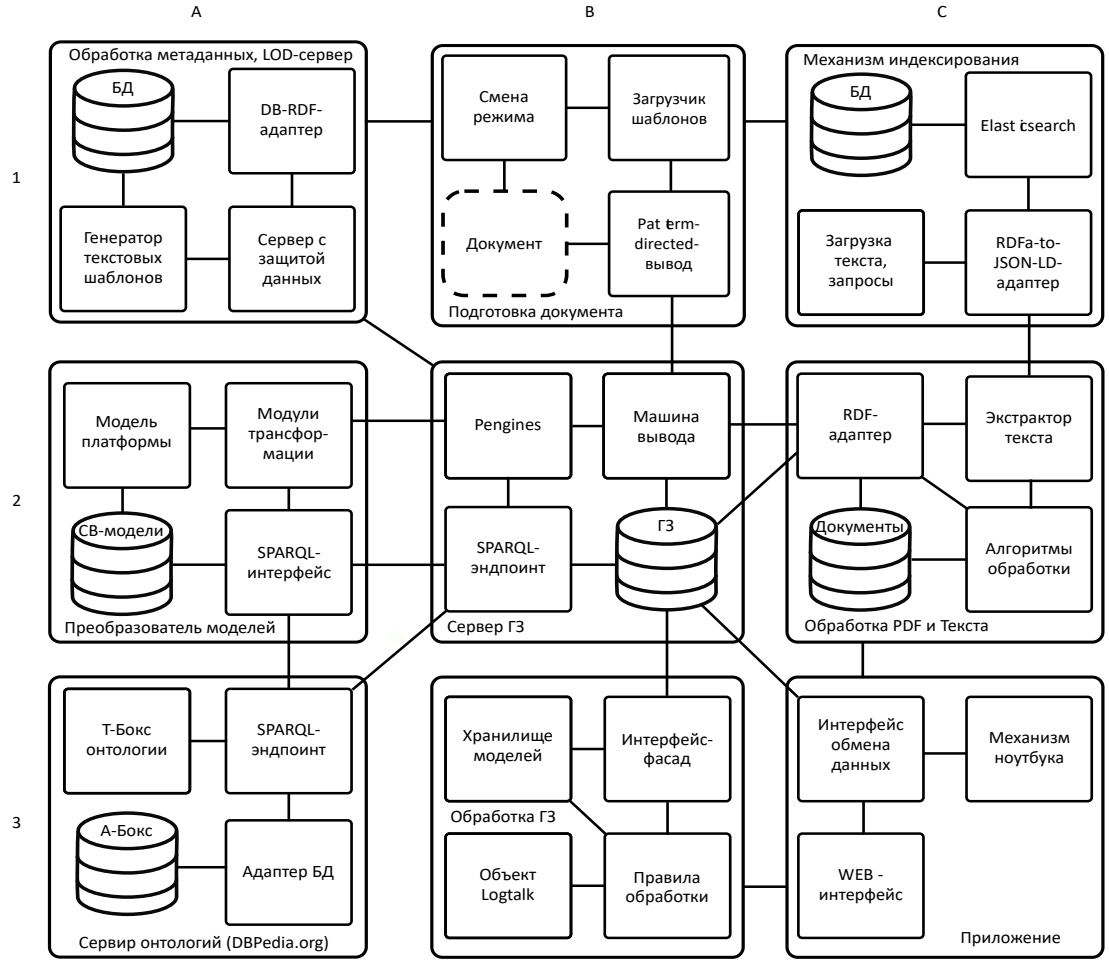

Рис. 1. Общая архитектура платформы разработки программного обеспечения на базе графа знаний

Разрабатываемые приложения строятся на основе доступа к серверам ГЗ (B2), веб-сервере с плагином Pengines [6]. Сервер ГЗ Virtuoso поддерживает UPDATE- и DELETE-запросы языка SPARQL, обеспечивая обновление данных, система масштабируема в локальной сети. Сервис Pengines позволяет веб-приложениям СВ использовать логический вывод на стороне сервера, интегрировать его со средой JavaScript браузера. База знаний Pengine композируется за счет программных модулей Prolog [7, 8] и модулей, передаваемых с JavaScript-приложения. Центральный ГЗ хранит общую часть A- и T-боксов данных приложений, модель предметной области. Другие модули и ресурсы среды оснащают функции конкретных приложений (C3).

Средства создания и интеграции данных документов (B1) – инструмент, позволяющий реализовать порождение контента документов при помощи технологий СВ [9], шаблонов, данных с других веб-страниц и сайтов, предоставляющих ресурсы LOD[1] [11, 12]. Созданный документ храниться в ГЗ и индексируется в модуле полнотекстовой индексации (C1). Совместно с Virtuoso он позволяет хранить и двоичные объекты в форматах, имеющих текстовый слой. Для модуля существуют две реализации. Одна построена на поисковой системе Sphinx, а вторая – Elasticsearch [13]. Elasticsearch использует JSON в качестве единственного

[1] Открытые связанные данные

формата представления документа. Данные ГЗ представимы в формате JSON-LD, что обеспечивает входные данные для индексирования. Другая реализация основывается Sphinx Search, она намного быстрее индексирует текст, потребляет меньше оперативной памяти, реализована в языке C++. Выбор механизма индексирования зависит от основного формата документа, используемого в приложении.

Объекты BLOB, хранящиеся в виде неструктурированных документов ГЗ (PDF-файлы, сканированные документы), как правило, содержат ценные данные, которые должны быть распознаны и проанализированы для использования в среде. Такими документами выступают данные отчетов, содержание научных статей, файлы DJVU и растровые изображения. Распознавание данных реализовано в модуле обработки PDF/текстовых документов (C2). Здесь производится распознавание и анализ страниц документа, результаты передаются на полнотекстовое индексирование, добавляются в текстовые слои в исходные документы и формируют структуры ГЗ (высокого порядка). Все полученные слои хранятся в базе данных C1, и данные, представляющие особый интерес, преобразуются в тройки ГЗ.

Обработка ГЗ общего характера выполняется в модуле B3. Этот компонент задает набор правил, используемых для реализации проверки, вывода эмерджентной семантики ГЗ, а также анализа и синтеза новых данных, включая целевой результат. Модуль A2 является источником данных об абстрактных моделях (моделей предметных областей, UML, формализаций постановок задач и т.п.), которые трансформируются и формируют Т-боксы ГЗ. Преобразование представляет вариант реализации архитектуры, управляемой моделью (MDA, Model Driven Architecture) [14]. Трансформация осуществляется под управлением модели платформы реализации, задающей контекст преобразования. Например, в случае синтеза исходного кода программного обеспечения подсистем модель платформы используется для реализации объектов приложений.

Модуль обработки метаданных (A1) оснащает среду возможностью задания или вывода семантики для выходных данных, порождаемых другими модулями обработки данных. Сервис позволяет сохранять характеристики выходных данных в ГЗ для дальнейшего использования в модуле (B3) логического вывода при автоматизации принятия решений. Модуль A3 обозначает собой внешние сервисы и ГЗ с ценными ресурсами, например, обозначениями глобальных объектов в DBPedia [15]. Эти сервисы являются основой поддержки средой требований LOD.

Обработка PDF-документов РПД. Входные данные в процедуру анализа – PDF-документы, представляющие РПД и загружаемые с веб-сайта ИГУ. Результат работы процедуры – распознанные содержательные данные, пригодные для императивной обработки. Обычно такой анализ возможно полноценно реализовать, поскольку PDF-файлы создаются из документов MS Word (не сканируются). Для РПД содержательной информацией является название и код курса, перечень тем, распределение зачетных единиц (академических часов) между лекциями, практикой, семинарами, самостоятельной работой студента, списком вопросов для оценки знаний и т.д.

Обработка начинается с преобразования исходного PDF в XML специального формата при помощи открытой библиотеки Poppler. Дерево XML обрабатывается объектами, реализованных в виде системы, основанной на знаниях, представленных на объектно-ориентированном логическом языке программирования Logtalk [16]. Структура исходных данных XML-дерева – это список страниц с последовательностями отрезков текста (run), строк и определений шрифтов. Списки конвертируются в базу данных объекта Logtalk, где каждому элементу задается номер для сохранения исходного порядка, соседние элементы одного типа (шрифт, страница, строка, кусок текста (КТ)) помечены специальным предикатом для организации быстрого доступа к рядом стоящим элементам.

Система распознает основные атрибуты (features) каждого КТ (последовательность символов, имеющих общий стиль шрифта) и строк, например, является ли строка висячей строкой абзаца, или присутствует ли число в начале строки. Для всех текстовых строк стра-

ницы определяются ограничивающий прямоугольник (bounding box), игнорируя номера страниц (одно- или двухсимвольные строки у верхнего или нижнего края страницы с номером, равным номеру страницы PDF). Полученные характеристики используются в объединении КТ и строк в абзацы, ограничивающий прямоугольник также используется для распознавания дополнительных свойств строк. Каждое объединение строк и КТ сопровождается пересчетом геометрических параметров абзаца. Абзацы, разделенные разрывами страниц, реконструируются на втором этапе, используя специальные правила.

Следующим этапом является распознавание структуры разделов РПД. Каждый институт ИГУ поддерживает свои шаблоны для создания РПД, так что последовательности разделов и стили шрифтов заголовков схожи внутри подразделения, но различаются между институтами. Заголовки распознаются по номерам разделов в исходном шаблоне, сверяясь с последовательностью корней слов, составляющих заголовок раздела. Вариации шаблона учитываются путем указания категории (category) Logtalk, задающей специализированные правила и корректирующей параметры общих правил. Затем, все строки/абзацы связываются с предшествующим заголовком. Заголовки и подзаголовки также находятся в иерархических отношениях. Для каждого заголовка в программе задана его семантика. В конце этапа документ распознан до древовидной структуры основных разделов документа и отдельных абзацев.

После построения иерархии запускается процесс распознавания нумерованных и ненумерованных списков. Здесь предполагается, что маркированные (ненумерованные) списки имеют более глубокий уровень вложенности в сравнении с нумерованными. Процесс состоит из двух этапов: обход всех структур, подобных списку, и поиск общих последовательностей чисел с постоянно увеличивающимися значениями или одинаковых маркировочных символов, пытаясь сохранить возможную вложенность; собственно само свертывание списков. При обнаружении однострочного списка, этот случай рассматривается как ложноположительный, строка присоединяется к предыдущему абзацу в виде обычной строки. Это правило распознает, например, номера страниц в ссылках на литературу, оказавшихся при верстке документа на отдельной строке. На этом этапе НИРОКР мы игнорируем списки описаний терминов, начинающиеся с существительного, за которым следует двоеточие или тире.

Последний этап – это экстракция требуемых данных о РПД из текста. На этом этапе очень важны данные о контекстах, задаваемых семантиками заголовков и структурами свертки списков. Контексты ограничивают набор абзацев, в которых расположены нужные данные. Местоположение данных указывается контекстом и списком предшествующих искомому значению корней слов и знаков препинания. Целевые данные извлекаются с помощью регулярных выражений или других процедур обработки строк.

Основным методами программирования объектов в подсистеме распознавания данных РПД, используемых при реализации анализа документов, являются расширение (extention) и композиция объектов. Исходная XML-структура документа инкапсулируются в параметрический объект Logtalk [16], параметром которого является имя временного файла, содержащего XML. Данный объект предоставляет базовые функции ввода-вывода, преобразование из дерева XML в базу данных объекта, изменение содержимого базы данных с сохранением согласованности, печать сообщений отладки и т.п. Объекты расширяются при помощи импорта набора категорий, реализующих этапы анализа и синтеза, конкретизирующих набор конфигурационных предикатов. В результате такой композиции создаются объекты, реализующие распознавание структуры конкретного типа документов.

Объект распознавания конструируется с учетом свойств конкретного шаблона РПД. Система импортированных категорий настраивается под шаблон при помощи объектов-потомков, улучшающих точность распознавания. По сравнению с существующими компонентными системами программирования в Logtalk конфигурация реализуется теми же программными структурами, что и реализация предикатов, все это благодаря абстрактному синтаксису Prolog/Logtalk. Более того, параметры конфигурации также могут быть правилами, выводящими значения из существующего контекста. Извлеченные данные сохраняются в

локальный ГЗ, который добавляется к основному ГЗ, хранящему данные учебных планов вуза.

Формирование и порождение PDF-документа РПД. Данные РПД, хранящиеся в ГЗ, помещаются в LuaLaTeX-шаблон [17], реализованный с использованием специального LaTeX-класса sucourse, подкласса KOMA-скрипта scrartcl. В классе sucourse реализованы специальные команды и блоки (environment) для определения распределения учебных единиц между лекциями, практиками и др. при помощи ключевых слов. Рассмотрим пример формирования перечня лабораторных работ, выраженных специальными LaTeX-блоками.

```
\begin{labworks}[comp={PC-4,PC-12}] % Компетенции по умолчанию,
                                    % удовлетворяемые перечнем лабораторных работ.
. . .                               % Предыдущие лабораторные работы
\begin{work}[comp={PC-4},           % компетенции данной лабораторной работы
    hours=16,                       % количество академических часов на работу
    topics={4},                     % Набор лекций с материалом для работы
    label=lw:distr]                 % Ссылки в терминологии LaTeX \label...
    {Разработка распределенной вычислительной среды}
                                    % Название лабораторной работы
    \paragraph{Problem definition:}Реализовать среду обработки данных
       типа «Map-Reduce» в горизонтальном кластере.
    \paragraph{Дано} Результаты предыдущих лабораторных работ.
    \paragraph{Разработать} Сеть взаимодействующих сервисов,
       реализующих потоковую (dataflow) архитектуру обработки информации.
    \paragraph{Дополнительное задание} Организовать управление вычислениями
       при помощи сервера RabbitMQ.
\end{work}
\end{labworks}
```

Лабораторные работы определяются двумя блоками labworks, настраивающим список-перечисление лабораторных работ и контекст RDF для сбора данных СВ; work, задающим конкретную лабораторную работу. Функции блоков в классе scrartcl задаются в так называемом новом расширенном синтаксисе LaTeX2e, позволяющем программисту управлять типами параметров определяемых элементов. Текущая альфа-версия класса sucourse опубликована на сайте Github [17].

```
\NewDocumentEnvironment{works}{O{}} % Определение блока в расширенном синтаксисе
{ % Запускается в точке \begin{...work}
  \def\itemname{Работа студента} % Заготовка названия работы студента
  \begin{syll@items}[#1]  % Обработка ключевых слов
  \begin{rdfctx}{\rdfsetctx{list}{syll wpdd:itemList % Разметка RDF
  !wpdd:ExampleList !wpdd:CurrentAttestation !wpdd:ItemList}}
  \def\syllabus@worktype{wpdd:LaboratoryWork} % Тип элемента списка в RDF
} % Конец части \begin...
{% Запускается в точке \end{...work}
  \luadirect{ syll.items:workValidation() } % Запуск проверки табличных данных
  \end{rdfctx} % Конец контекста RDF
  \end{syll@items} % Конец блока
}

\NewDocumentEnvironment{work}{O{}m} % Задать работу с обязательным заголовком
{
  \begin{syll@item}[#1]{#2} % Setup item with a parent assets
  \begin{rdfenv}{list ^schema:member !wpdd:ListItem
     !wpdd:Example \luadirect{self.item:sprintRDFTypes()} }
  \paragraph{{\workheaderstyle \itemname\ \theitem.}~{\worktitlestyle #2}}
```

```
    % Формат начала элемента списка
}
{
  \end{rdfenv}\par\vspace{1em} % Добавить пустую строку между лабораторными
  \end{syll@item}
}

\NewDocumentEnvironment{labworks}{O{}} % Специализация блока «works»
{
\syl@labwork@section % Начать с адаптера общего шаблона РПД
\begin{works}[totalnames={'hours'},
  type=labwork,      % Разновидность академического часа (лекция, лабораторка
  itemname={Лабораторная работа},  % Задать имя элемента списка
  rdftype={LaboratoryWork}, #1]}% Задать тип RDF для элемента
{\end{works}}
```

Блоки неявно выполняют Lua-код для сбора данных, проверки ограничений и генерации таблиц и других LaTeX-структур РПД. Если логические ограничения не выполняются, в исходный код LaTeX РПД, транслируемый LuaLaTeX, добавляется сообщение об ошибке в виде текста, окрашенного в красный цвет. Проверку можно отключить при помощи ключевого слова final в команде задания класса LaTeX-документа (ключевое слово draft включает отображение результатов проверки целевой PDF, даже если все заполнено правильно). Код Lua также генерирует вспомогательные файлы с данными, обрабатываемыми между запусками программы lualatex.

Данная функция используется для проверки наличия печатных изданий в библиотеке университета, обновления и проверки существования URL-ссылок электронных изданий. Функция реализована асинхронно при помощи RabbitMQ и другими сервисами загрузки и анализа библиографических записей в системах ИРБИС ИГУ, серверах Лань и им подобных. Для ускорения обработки РПД в одном из подграфов ГЗ хранятся данные обработанных библиографических ссылок изданий. Собранные Lua-скриптами данные УП также отправляются в KG.

Использование LuaLaTeX как основной механизм обработки текста определяется его выдающимися возможностями по качеству верстки документов, высокоуровневой разметкой, командами и блоками, доступностью шрифтов True-type и поддержкой Lua, позволяющего расширять функции LaTeX и взаимодействовать с параллельными процессами. Специализированные команды класса sucourse не дают пользователю изменять стили. Определением новых полезных структур можно поддерживать создание документов с общими текстовыми структурами, представляющими различные аспекты учебной программы. Типичным примером являются тесты, интегрируемыми в исходный текст, а затем экспортируемыми в формат, принимаемый системой Moodle. Класс настраивается на свойства шаблона РПД при помощи импорта файла настройки по имени шаблона (параметр класса).

Начальное состояние LaTeX-исходника РПД генерируется программой Python и механизмом шаблонизации Jinja, расширенного функциями поддержки RDF. Jinja расширен синтаксическими структурами, позволяющими делать запросы к основному ГЗ или его локальный подграф для получения необходимых данных. Синтаксические расширения реализуют запросы, возвращающие значения некоторого типа (rdf:type), блоки обработки наборов ответов на запрос и определения RDF-контекстов. Получаемый проект документа также включает семантическую разметку, невидимую в окончательном PDF, но позволяющую экстрагировать данные RDF из исходных текстов LaTeX РПД (аналогично RDFa в HTML).

Ручное редактирование проекта – этап, выполняемый преподавателем, подготавливающим РПД. На этом этапе в документ вносятся новые разделы, и, при условии сохранения сгенерированной разметки, при компиляции tex-файла исходника РПД данные РПД передадутся в ГЗ.

Реализации-аналоги. В ходе поиска аналогов и работы в сотрудничестве с другими вузами Иркутска и Санкт-Петербурга, работы с on-line-издательствами, мы собрали данные о характеристиках используемых ими программных продуктов, релевантных, нашей задаче.

Санкт-Петербургский электротехнический университет (СПбГЭТУ) разработал систему автоматизации подготовки текстов учебных программ [18]. Система позволяет преподавателям вводить значимую информацию о РПД и др. документов в хранилище в формате JSON при помощи заполнения форм. РПД генерируются в виде исходного кода LaTeX и транслируется в PDF. Секретарь кафедры получает PDF-файлы и публикует их на домашней странице курса. Система постоянно развивается в специальном отделе университета. К настоящему времени реализован документооборот между преподавателями и контролирующими подразделениями, определены соответствующие роли. Состояние обработки документа представлено в пользовательском интерфейсе. Система способна хранить произвольные PDF-документы и генерировать полный пакет документов для представления органам квалификации (ученый совет, министерство).

Менее совершенный генератор разработан в Национальном исследовательском государственном техническом университете (ИрНИТУ) [19]. Серверное PHP-приложение предоставляет данные учебных планов, сгенерированные программой «Шахты». Пользовательский интерфейс преподавателя позволяет определять формальные части РПД, такие, как темы лекций, список лабораторных работ, самостоятельных работ и семинаров с распределением зачетных единиц между ними. На заключительном этапе генерируется документ Word в формате DOCX. Данные, не имеющие отношение к УП, должны заполняться вручную ежегодно: список тестов, вопросы к экзамену, ссылками на литературу и т.п.

Поиск в Google в российской части интернета приводит к большему количеству примеров генераторов РПД, грубый обзор предоставляемых функций в руководствах пользователя выявляет высокую заинтересованность в развитии средств автоматизации подготовки РПД и др. документов. Как правило, это системы, аналогичные представленным выше.

Достаточно простой вариант порождения частей РПД реализован в виде подсистемы в ИС «Лань» [20]. Вариант ориентирован только на автоматизацию сбора электронных изданий для РПД. ИС хранит все книги издательства, структурированные по множеству тегов предметной области, и предоставляет списки литературы, отфильтрованные по схожим темам, относящимся к набору ранее выбранных экземпляров. Из списка в автоматическом режиме формируется раздел РПД.

В нашем НИРОКР планируется создать систему, использующую существующие структурированные и неструктурированные данные для построения информационной модели образовательного процесса ИМИТ ИГУ, а также масштабирование результатов на другие факультеты.

Заключение. На данном этапе НИРОКР разработан набор модулей (сервисов) поддержки анализа информации, размещенной в PDF-документах на сайте вуза о структуре преподаваемых курсов (рабочих программ дисциплин, РПД). Сервисы функционируют независимо друг от друга, сохраняя информацию в распределенном графе знаний, технологии Семантического Веба. Такая архитектура позволяет интегрировать приложения и разрабатывать новые достаточно независимо друг от друга.

Функционирование модулей инфраструктуры основана на активном использовании метаданных, включая распределенные и удаленные ресурсы семантических данных, что призвано обеспечить общую модель предметной области при проектировании структур данных.

Среди успешных результатов класса MVP выделены два подпроекта, представленных статье:

- распознавание структуры данных РПД из PDF-документов и
- порождение РПД с использованием данных в ГЗ.

Дальнейшее развитие предполагается вести в двух основных направлениях:

- повышение качества распознавания РПД за счет автоматизации распознавания таблиц PDF-документов алгоритмами, реализованными коллегами [21, 22],
- интеграция данных РПД с программами планирования нагрузки, составления расписания.

Благодарности. Результаты получены в рамках государственного задания Минобрнауки России, проекта «Методы и технологии облачной сервис-ориентированной цифровой платформы сбора, хранения и обработки больших объемов мультиформатных междисциплинарных данных и знаний на основе использования искусственного интеллекта, модельно-ориентированного подхода и машинного обучения», No. FWEW-2021-0005. (Государственная регистрация No 121030500071-2). В проекте использована сетевая инфраструктура Телекоммуникационного центра коллективного пользования «Интегрированная информационно-вычислительная сеть Иркутского научно-образовательного комплекса» (http://net.icc.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Stojanov Z., Stojanov J., Jotanovic G., Dobrilovic D. Weighted networks in socio-technical systems: Concepts and challenges // CEUR-WS Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments Irkutsk, Russia, July 6-7 – 2020 – pp. 265–276.

2. Hogan A., Blomqvist E., Cochez M., D'Amato C. *et al*. Knowledge Graphs – 2020 – URL:https://arxiv.org/abs/2003.02320v5 (access date: 12-Dec-2021)

3. Erling O. Virtuoso, a Hybrid RDBMS/Graph Column Store // IEEE Data Eng. Bull. 2012. vol. 35 – pp. 3–8.

4. J. Wielemaker, W. Beek, M. Hildebrand, J. Ossenbruggen, ClioPatria: A SWI-Prolog infrastructure for the Semantic Web // Semantic Web – vol. 7(5) – pp. 529–541 (2016)

5. Berners-Lee T., Hendler J., Lassila O. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities // Scientific American, May 2001.

6. Lager T., Wielemaker J. Pengines: Web Logic Programming Made Easy. Theory and Practice of Logic Programming – vol. 14, no. 4-5 – 2014 – pp. 539–552.

7. Wielemaker J., Schreiber G., Wielinga B. Prolog-based infrastructure for RDF: scalability and performance // In: D. Fensel, K. Sycara, J. Mylopoulos (eds) The Semantic Web – ISWC 2003. ISWC 2003. Lecture Notes in Computer Science. – vol. 2870 – Springer, Berlin, Heidelberg, 2003.

8. Wielemaker J., Schrijvers T., Triska M., Lager T. SWI-Prolog // Theory and Practice of Logic Programming – vol. 2, no. 2 – 2011 – pp. 67–96 – ISSN 1471-0684.

9. Cherkashin E., Shigarov A., Paramonov V., Mikhailov A. Digital archives supporting document content inference // Procs. of 42-nd International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), May 20–24, Opatia, Chroatia – 2019 – pp. 1037-1042.

10. Cherkashin E., Shigarov A., Paramonov V. Representation of MDA transformation with logical objects // Procs. of International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON) – Novosibirsk, Russia, 2019 – pp. 0913–0918

11. Bizer Ch., Heath N., Berners-Lee T. Linked data – the story so far, International // Journal on Semantic Web and Information Systems – vol. 5 (3) – 2009 – pp. 1–22.

12. Heino N., Tramp S., Heino N., Auer S. Managing web content using linked data principles – combining semantic structure with dynamic content syndication // Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual – pp. 245–250 – URL: http://svn.aksw.org/papers/2011/COMPSAC_lod2.eu/public.pdf

13. Kuć R., Rogoziński M. Mastering Elasticsearch – Second edition, Packet Publishing. 2015 – 372 p.

14. Cherkashin E., Terehin I., Paramonov V. New transformation approach for Model Driven Architecture // Proceedings of the 35th International Convention MIPRO, Opatija, Choatia, 2012 – pp. 1082-1087.

15. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., *et al.* DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia // Semantic Web Journal – OS Press, 2015 – vol. 6, No. 2 – pp. 167–195

16. Moura P. Programming Patterns for Logtalk Parametric Objects // In: Abreu, S., Seipel, D. (eds) Applications of Declarative Programming and Knowledge Management. INAP 2009. – Lecture Notes in Computer Science – Springer, Berlin, Heidelberg – 2011 – vol. 6547

17. Класс LuaLaTeX для оформления учебных программ вуза [Сайт]. URL:https://github.com/eugeneai/sucourse (дата доступа: 10.10.2022)

18. СПбГЭТУ “ЛЭТИ” [Сайт]. URL:https://etu.ru/ (дата доступа: 10.10.2022)

19. ИРНИТУ – университет с лучшими традициями [Сайт]. URL:https://istu.edu/ (дата доступа: 10.10.2022)

20. Библитечная система Лань [Сайт]. URL:https://lanbook.com/ (дата доступа: 10.10.2022)

21. Shigarov A., Paramonov V., Belykh P., Bondarev A. Rule-based canonicalization of arbitrary tables in spreadsheets // In: Dregvaite G., Damasevicius R. (eds) Information and Software Technologies. ICIST 2016.

22. Shigarov A., Mikhailov A. Rule-based spreadsheet data transformation from arbitrary to relational tables // Information Systems – vol. 71 – 2017.

REFERENCES

1. Stojanov Z., Stojanov J., Jotanovic G., Dobrilovic D. *Weighted networks in socio-technical systems: Concepts and challenges.* CEUR-WS Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments Irkutsk, Russia, July 6-7, 2020, pp. 265–276.

2. Hogan A., Blomqvist E., Cochez M., D’Amato C. *et al*. *Knowledge Graphs.* 2020, URL:https://arxiv.org/abs/2003.02320v5 (access date: 12-Dec-2021)

3. Erling O. *Virtuoso, a Hybrid RDBMS/Graph Column Store.* IEEE Data Eng. Bull, 2012. vol. 35 – pp. 3–8.

4. Wielemaker J., Beek W., Hildebrand M., Ossenbruggen J. *ClioPatria: A SWI-Prolog infrastructure for the Semantic Web.* Semantic Web. 2016, vol. 7(5), pp. 529–541

5. Berners-Lee T., Hendler J., Lassila O. *The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, May 2001.

6. Lager T., Wielemaker J. *Pengines: Web Logic Programming Made Easy*. Theory and Practice of Logic Programming. 2014, no. 4-5, vol. 14, pp. 539–552.

7. Wielemaker J., Schreiber G., Wielinga B., *Prolog-based infrastructure for RDF: scalability and performance.* In: D. Fensel, K. Sycara, J. Mylopoulos (eds) The Semantic Web – ISWC 2003. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2003, vol. 2870.

8. Wielemaker J., Schrijvers T., Triska M., Lager T. *SWI-Prolog*. Theory and Practice of Logic Programming, 2011, no. 2, vol. 2, pp. 67–96, ISSN 1471-0684.

9. Cherkashin E., Shigarov A., Paramonov V., Mikhailov A. *Digital archives supporting document content inference.* Procs. of 42-nd International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), May 20–24, 2019, pp. 1037-1042.

10. Cherkashin E., Shigarov A., Paramonov V. Representation of MDA transformation with logical objects. Procs. of International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON) Novosibirsk, Russia, 2019, pp. 0913–0918

11. Bizer Ch., Heath N., Berners-Lee T., *Linked data – the story so far*. International Journal on Semantic Web and Information Systems. 2009, vol. 5 (3), pp. 1–22.

12. Heino N., Tramp S., Heino N., Auer S., *Managing web content using linked data principles – combining semantic structure with dynamic content syndication*. Computer Software and Applications Conference (COMPSAC), IEEE 35th Annual, 2011, pp. 245–250. URL:http://svn.aksw.org/papers/2011/COMPSAC_lod2.eu/public.pdf

13. Kuć R., Rogoziński M. *Mastering Elasticsearch - Second edition, Packet Publishing*. 2015, 372 p.

14. Cherkashin E., Terehin I., Paramonov V. *New transformation approach for Model Driven Architecture.* Proceedings of the 35th International Convention MIPRO, Opatija, Choatia, 2012, pp. 1082-1087.

15. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., *et al*, *DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia.* Semantic Web Journal. IOS Press. 2015, no. 2, vol. 6 pp. 167–195

16. Moura P. *Programming Patterns for Logtalk Parametric Objects.* In: Abreu, S., Seipel, D. (eds) Applications of Declarative Programming and Knowledge Management. INAP-2009. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2011, vol. 6547.

17. A LuaLaTeX class for authoring course description. URL:https://github.com/eugeneai/sucourse (access date: 10.10.2022)

18. ETU “LETI”. URL:https://etu.ru/en/university/ (access date: 10.10.2022)

19. INRTU is a university with the best traditions… URL:https://eng.istu.edu/ (access date: 10.10.2022)

20. LMS Lan’. URL:https://lanbook.com/ (in Russian) (access date: 10.10.2022)

21. Shigarov A., Paramonov V., Belykh P., Bondarev A. *Rule-based canonicalization of arbitrary tables in spreadsheets.* In: Dregvaite G., Damasevicius R. (eds) Information and Software Technologies. ICIST 2016.

22. Shigarov A., Mikhailov A. *Rule-based spreadsheet data transformation from arbitrary to relational tables.* Information Systems, 2017, vol. 71.

Информация об авторах

Евгений Александрович Черкашин – канд. техн. наук, старший научный сотрудник Института динамики систем и теории управления им. В.М. Матросова СО РАН, зав. каф. информационных технологий института математики и информационных технологий Иркутского государственного университета, г. Иркутск, e-mail: eugeneai@icc.ru

Виктория Алексеевна Попова – аспирант, ассистент кафедры алгебраических и информационных систем Института математики и информационных технологий Иркутского государственного университета, г. Иркутск, e-mail: victorypopova1@gmail.com

Authors

Evgeny Alexandrovich Cherkashin – Ph. D. in Engineering Science, Senior Researcher at Matrosov Institute for System Dynamics and Control Theory SB RAS, Head of Chair of Information Technology, Institute of Mathematics and Information Technology, Irkutsk State University, Irkutsk, e-mail: eugeneai@icc.ru

Victoria Alekseyevna Popova – Postgraduate student, Assistant of the Department of Algebraic and Information Systems, Institute of Mathematics and Information Technologies, Irkutsk State University, Irkutsk, e-mail: victorypopova1@gmail.com

Для цитирования

Черкашин Е.А., Попова В.А. Распределенная инфраструктура для обработки документов образовательного процесса, основанная на графах знаний // «Информационные технологии и математическое моделирование в управлении сложными системами»: электрон. науч. журн. – 2022. – №4(16). – С. 44-55 – DOI: 10.26731/2658-3704.2022.4(16).44-55 – Режим доступа: http://ismm-irgups.ru/toma/416-2022, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 21.12.2022).

For citations

Cherkashin E.A., Popova V.A. Knowledge graph based distributed infrastructure for processing education process documents // Informacionnye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami: ehlektronnyj nauchnyj zhurnal [Information technology and mathematical modeling in the management of complex systems: electronic scientific journal], 2022. No. 4(16). P. 44-55. DOI: 10.26731/2658-3704.2022.4(16).44-55 [Accessed 17/12/22].